基于注册码的软件授权保护系统的设计与实现

何永瑾[1,2],郭肖旺[2],赵德政[2]
(1.华北计算机系统工程研究所,北京 100083; 2.中电智能科技有限公司,北京 102209)

摘　要:目前国内软件加密授权技术的发展较为缓慢,针对传统的软件保护方式单一,软件授权方式不灵活的问题,提出了基于注册码的软件授权保护方案,设计了软件的加密授权和检验流程,对软件加密授权的各环节进行优化和改进。该方案不仅为用户提供了软件的授权保护机制,如时间授权、功能授权等,还能在离线环境下增强软件的保护作用,具有较高的安全性和实用性。

关键词:软件保护;软件授权;SM4 算法
中图分类号:TP311.5　　**文献标识码**:A　　DOI: 10.19358/j.issn.2096-5133.2020.05.009
引用格式:何永瑾,郭肖旺,赵德政.基于注册码的软件授权保护系统的设计与实现[J].信息技术与网络安全,2020,39(5):42-45,50.

Design and implementation of software authorization protection model based on registration code

He Yongjin[1,2], Guo Xiaowang[2], Zhao Dezheng[2]
(1. National Computer System Engineering Research Institute of China, Beijing 100083, China;
2. Intelligence Technology of CEC Co., Ltd., Beijing 102209, China)

Abstract: At present, the development of software encryption and authorization technology in China is relatively slow. Aiming at the problems of single traditional software protection mode and inflexible software authorization mode, this paper proposes a software authorization protection scheme based on registration code, designs the software encryption authorization and inspection process, and optimizes and improves each link of software encryption authorization. This scheme not only provides users with the authorization protection mechanism of the software, such as time authorization, function authorization, etc., but also enhances the protection function of the software in the offline environment, with high security and practicability.

Key words: software protection; software authority; SM4

0　引言

计算机软件的发展和应用为社会带来了巨大的效益,也提供了大量的就业岗位。在为人们带来便利的同时,软件保护的问题也开始备受关注。由于软件厂商版权意识不强,人们对知识产权不够重视,盗版软件和非授权软件的使用不仅使企业遭受经济损失,更容易使用户和软件的信息泄露,被不法人员利用。2018 年,商业软件联盟(BSA)公布了《全球软件调查》报告,报告结果表明,我国计算机软件盗版率同 2014 年相比下降了 8 个百分点。这一方面归功于国家一直在完善的软件知识产权保护制度,另一方面软件研发者们不断研究各种各样的软件保护技术应用在自己的软件上,延长非法破解软件时间,增加不法分子盗版软件的技术和时间成本,尽可能保护软件不被非法利用[1]。

按照工作方式和原理的不同,国内外常见的软件保护技术分为两种,一种是基于软件的软加密技术,一种是需要特定硬件配合的硬加密技术。然而不论是软加密还是硬加密,都有不可避免的劣势。硬加密的缺点在于:①容易引起硬件冲突,操作要求高;②随着软件的升级,加密锁也需要不断更换,成本高;③需要附加专业硬件设备,灵活性差。软加密技术减少了硬加密的一些缺点,不需要附加硬件设备,不过软加密使用的 keyfile 和序列号的格式大同小异,容易被篡改,无法保证安全性。

在国外,软加密保护产品凭借其易分发、成本

低、灵活性好等优势，已经代替加密锁成为主流趋势，如已经成熟的 Flexlm 系统，软许可 CmActLicense 等。目前国内加密锁仍是主流软件保护产品，自主研发的软加密授权产品较少，系统仍有待完善[2]。

国内的软件保护技术目前存在的问题在于现有的软件保护的方式过于单一，如硬件绑定、加壳技术等，未对软件的保护方法进行二次设计开发，导致软件破解有规律可循；再者，软件多侧重于对软件自身的保护，购买一次可以获得永久的使用权，没有结合用户的需求采取灵活的授权模式，例如不同的用户应该拥有不同的使用期限和不同的功能模块。

本文提出了一种基于机器注册码的软件授权保护策略，优先使用国产加密算法，对软件授权的各个环节进行优化和改进。在保护软件的同时，也可以按时间、功能模块对用户进行授权。

1 系统方案

为解决传统软件保护技术存在的保护方式单一、授权不够灵活等问题，设计基于注册码的软件授权保护方案，该方案的设计目标包括以下四点。

(1)激活码需要和硬件信息相结合。硬件信息是唯一且不变的，做到“一机一码”，保证每台机器的激活码不可通用，单独授权，并且要做到硬件信息不易推测出来。

(2)生成激活码的算法的安全性。基于软加密的保护机制的核心在于加解密算法，采用安全可靠的加解密算法防止软件轻易被破坏。

(3)安全的授权校验机制，解密的过程是一个明文信息暴露的过程，如果校验机制出现了隐患，对软件来说则是致命的威胁。

(4)具有一定的自身检查能力，包括对磁盘文件和内存映像的检查，避免软件在开发者不知情的情况下修改校验机制，被破解而不自知。

基于上述设计目标，制定了软件授权保护系统的授权流程，其流程如图 1 所示。主要思路是：首先，客户启动软件，软件提取能够唯一标识机器身份的指纹信息，使用加密算法对其加密形成注册码。服务端收到注册码之后，根据软件的基本信息和用户的需求信息(功能需求和使用期限)生成激活码，发送给客户端。客户端使用激活码激活软件，在校验激活信息无误后，进入软件主界面。

2 系统组成部分

基于功能的需求，对系统功能进行功能模块的划分，得出系统功能组成，如图 2 所示。

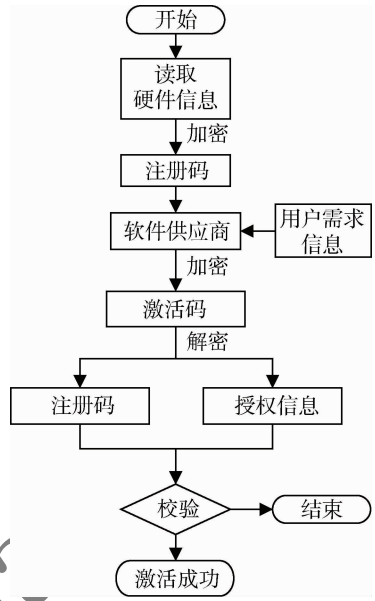

图 1 授权流程

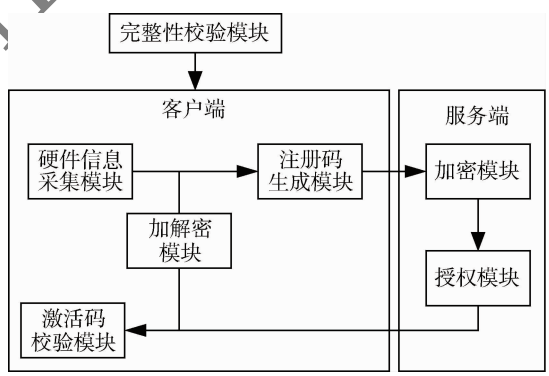

图 2 系统组成

(1)服务端的加密模块和授权模块是本系统的重要功能模块。授权模块需要读取客户端发来的注册码和用户的需求信息并调用加密模块的加密算法对信息加密。

(2)客户端的加解密模块包括生成注册码和完整性校验时需要的加密摘要算法，以及解析激活码时的解密算法。

(3)硬件信息采集模块主要用于采集可以唯一标识用户身份的硬件信息，如磁盘序列号、MAC 地址等。

(4)注册码生成模块包括读取硬件信息、组合硬件信息和生成信息摘要三部分。

(5)激活校验模块获取到解密后的信息后校验内容的正确性,包括激活码校验、时间校验和功能校验。

(6)完整性校验模块在程序最开始运行的时候对整个软件的完整性进行检验,校验无误后才可进入激活校验。

3 系统具体实现

3.1 注册码生成模块

计算机中,可以作为指纹的能读取的硬件信息有:MAC 地址,CPU 序列号,BIOS 序列号,硬盘序列号。MAC 地址具有全球唯一性;BIOS 序列号的读取并不容易,但有些主板没有 BIOS 序列号,使用一些工具也可以轻易更改它,如 DMIScope;如果是同一批次同一配置的电脑,CPU 序列号存在重复的可能;硬盘序列号是独一无二的,是计算机信息的重要组成部分,如果执意修改,可能会导致系统错误,硬盘无法使用。所以本方案采取读取硬盘序列号和 MAC 地址作为指纹信息。采用这一方式有一定的风险,两种信息的简单组合很容易被推测出所使用的硬件信息,在安全性上有所欠缺。故本方案中将硬盘序列号和 MAC 地址分组进行交叉,使信息变得复杂化,如图 3 所示。

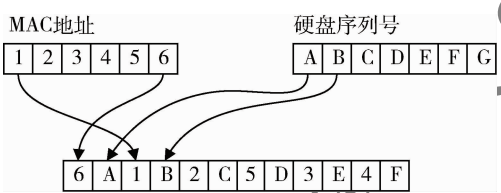

图 3 硬件信息组成方式

由于硬件信息的长度不固定,注册码的生成采用摘要算法,生成固定长度的消息摘要。MD5 和 SHA-1 算法不再安全[3],本文采用安全性更高的 SM3 摘要算法生成摘要。将其作为加密算法的消息输入,摘要算法加密的不可逆性,可以保证硬件信息的交叉处理原文不被反推出来。

3.2 加密授权模块

和非对称加密算法相比,对称密码算法运算速度快,加密效率高,不需要考虑安全传输的问题。在对称密码算法中,国产 SM4 对称分组密码算法具有较高的可靠性和安全性,故本方案中采用 SM4 算法。它的分组长度为 128 bit,密钥长度也为等长的 128 bit。加密算法和密钥扩展算法均采用 32 轮非线性迭代结构,每次迭代为一个圆形变换函数 f。该 SM4 算法添加/解密相同的结构,只使用车轮密钥相反,其中解密车轮密钥是加密车轮密钥的相反顺序[4]。

3.2.1 SM4 算法的改进使用

由于 SM4 是对称加密密钥,加密和解密过程使用相同的密钥,获取了加密密钥,相当于得到了解密密钥,就可以破解密文。另外 SM4 算法在扩展密钥的计算过程中使用固定参数 CK 和 FK,不具有良好的随机性。故在使用过程中,对 SM4 算法进行了改进[5]。本文提出在轮密钥的生成过程中添加随机性参数,改进参数的选择方式,无法获得这些参数,则无法生成正确的轮密钥,也无法还原解密过程。具体改进描述如下:

(1)将 4 个系统参数 FK 增加至 16 个,原来的 4 个系统参数不变,另外使用随机数生成 12 个不同的参数 $FK_4 \sim FK_{15}$。采取以下方式随机选择 4 个参数进行加密运算。

采用线性探测法构建哈希表,存储 16 个 FK,如图 4 所示。

FK_{10}	FK_7	FK_{13}	FK_2	FK_3	FK_6	FK_1	FK_{14}	FK_4	FK_{15}	FK_5	FK_{12}	FK_9	FK_0	FK_8	FK_{11}

图 4 固定参数表

取明文的最后一位 X_{15},模 4 求余得 $j = X_{15} \bmod 4$,计算取 FK 的关键字地址 $h_i = 4 \times i + j$ $(i = 0,1,2,3)$,得到 4 个 $FK_i = \mathrm{Hash}(h_i)(i = 0,1,2,3)$。

(2)将 32 轮密钥扩展和轮函数计算的过程增加到 36 轮,同时原来使用的固定参数 CK 不变,根据增加的轮数增加 4 个 CK 值。设为 CK_i 的第 j 字节 $(i = 0,1,\cdots,31; j = 0,1,2,3)$,即 $CK_{i,j} = (CK_{i,0}, CK_{i,1}, CK_{i,2}, CK_{i,3}) \in (Z_2^8)^4$,则:

$CK_{32} = 80878e95$ $CK_{33} = acb3bac1$

$CK_{34} = c8cfd6dd$ $CK_{35} = e1ebf2f9$

在固定参数 FK 中添加随机参数,增加了生成轮密钥的随机性,提高密钥的安全性。增加 CK 的值,并且增加密钥扩展的轮数,提高了算法的复杂度,进而提高整个算法的安全。

3.2.2 加密授权方法

激活码由软件供应商提供生成。SM4 算法为分组为 128 bit 长度的对称加密算法,加密数据格式较为固定,若一次加密,形式过于单一,故采用两次加密,先验证注册码的合法性,如果合法,则设置软

件信息,包括软件版本信息、注册时间等,合成等长的 128 bit 字符串作为密钥,对注册码进行第一次 SM4 加密。把授权信息,如使用期限、使用的功能、供应商等,生成 128 bit 的字符串,使用上一次生成的密钥,对授权信息进行第二次加密,生成最终的激活码[6]。生成激活码流程如图 5 所示。

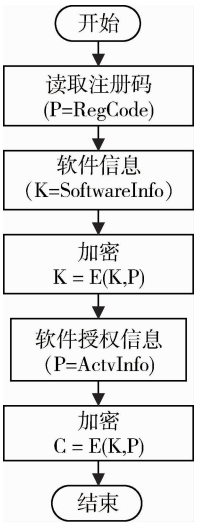

图 5　生成激活码流程

3.3　软件激活校验模块

验证激活码是为了判断用户是否有合法的使用权。验证过程在客户端进行,可以在无网络的环境下验证,保证其安全性。主要流程[7]如图 6 所示。

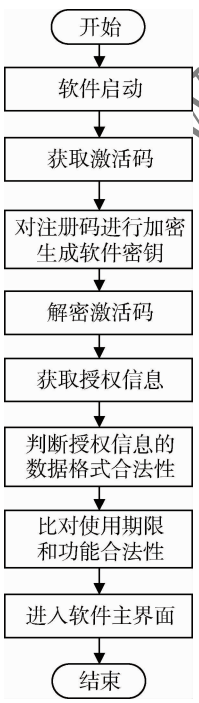

图 6　校验流程

获取到激活码后,首先和生成过程类似,将注册码和软件信息加密,生成解析授权信息的软件密钥,使用软件密钥对激活码解密,获取授权信息,判断授权信息的数据格式是否合法。这一步也是对激活码进行校验,如果数据格式不合法说明激活码中加密的注册码有误,无法对软件激活。获取到软件信息后,判断软件的使用期限和功能信息是否有效,有效则进入软件主界面。

在软件运行期间,会一直读取激活码,检查硬件信息是否相符。还会进行功能和时间校验。读取上次使用的时间,并计算出剩余使用时间,写入注册表中。在软件校验时间时,读取注册表中的时间信息,判断时间是否在有效期内,如果不是则标志为过期,阻止用户通过篡改时间达到延长期限的目的,保证软件一直运行在授权期限中[8]。如果在软件运行中修改时间,程序检测到当前时间大于上次使用时间,则停止运行,直到时间修改回来才能继续运行。

在客户端软件中嵌入功能模块信息列表,用户使用某一功能模块的时候,检测程序中的功能模块码值和授权信息中的功能信息是否一致,则开放对应模块的功能。

3.4　文件完整性校验

在保护方案中,为了阻止恶意程序修改软件信息,保护授权校验机制,增加了对软件的完整性校验[9]。完整性校验包括磁盘文件和内存映像的检查,DLL 文件和 EXE 文件的完整性检查。对原始文件使用 SM3 计算得到散列值,并将值放在某处,每当 EXE 文件开始运行时,重新计算文件的散列值,同原值进行比较,如果不同则表明文件已经被修改。对内存映像检测的方法是从内存映像中获得代码区块的 RVA 值和内存大小,计算内存数据的散列值,与自身文件先前储存的散列值进行比较,判断数据是否被修改。如果检测到数据被修改,则停止程序的运行。

4　结论

本文从实际应用出发,分析常见软件保护技术的不足之处,将软件保护和用户授权相结合,提出了一种基于注册码的软件授权保护方案,改进了注册码的构成方式,设计了授权和加密方案,完成了激活校验的流程,并对软件做了完整性校验,保证

(下转第 50 页)

究[D]. 长春:长春理工大学,2017.
[8] PAPA U, ARIANTE G, DEL CORE G. UAS aided landing and obstacle detection through LIDAR-Sonar data[C]. Proceedings of the 5th IEEE International Workshop on Metrology for AeroSpace, Rome, Italy, 2018.
[9] LI Y, HU D D, CHEN J X, et al. Research on obstacle avoidance algorithm for four-rotor UAV[C]. Proceedings of the 2018 2nd International Conference on Artificial Intelligence, Automation and Control Technologies, Osaka, Japan, 2018.
[10] 倪旭翔,胡凯. 脉冲串互相关方法在远程激光测距中的应用[J]. 光学学报,2012,32(11):128-133.

(收稿日期:2020-02-08)

作者简介:

潘世光(1995 -),男,硕士研究生,主要研究方向:激光雷达避障。

尚建华(1983 -),通信作者,女,博士研究生,副教授,主要研究方向:激光遥感、激光多普勒振动测量技术。E-mail:jhshang@ dhu. edu. cn。

罗远(1989 -),男,博士研究生,主要研究方向:激光三维成像技术和激光测距技术。

(上接第 45 页)

软件的完整性,在对软件起到保护作用的同时,可以对用户分时间和功能授权,提高了授权的灵活性。

本方案还可以做一些改进,例如增加云授权、次数授权等。还应该从不同角度进行安全防护,从底层系统的安全性开始同软件保护技术结合起来,保证软件的安全,这也是接下来重点研究的内容。

参考文献

[1] 段志秀. 计算机软件的著作权保护研究[D]. 兰州:兰州大学,2019.
[2] 余彦. 基于序列号的软件保护模型改进研究[D]. 兰州:兰州大学,2015.
[3] WANG X, YU H. How to break MD5 and other Hash functions[J]. Lecture Notes in Computer Science, 2005, 3494:561.
[4] 吕述望,苏波展,王鹏,等. SM4 分组密码算法综述[J]. 信息安全研究,2016,2(11):995-1007.
[5] 刘海峰,朱婧,曹慧. 改进 DES 子密钥使用顺序的算法研究[J]. 西南大学学报(自然科),2017,39(6):135-140.
[6] 欧阳雪,周寰,邓锦洲,等. 一种面向软件生命周期的 m 授权保护系统设计与实现[J]. 计算机工程与科学,2013,35(4):59-64.
[7] 赵舒灿. 一种软件授权与保密系统设计[J]. 电脑编程技巧与维护,2017(10):5-9,17.
[8] 李志龙. 基于非对称加密算法的软件授权方案[J]. 福建电脑,2018,34(9):116-118.
[9] 王健. 基于完整性验证和壳的软件保护技术研究[D]. 太原:中北大学,2018.

(收稿日期:2020-03-16)

作者简介:

何永瑾(1995 -),女,硕士研究生,主要研究方向:工业控制软件、工控安全。

郭肖旺(1986 -),女,硕士,工程师,主要研究方向:工业控制软件、工控安全。

赵德政(1985 -),男,博士,高级工程师,主要研究方向:工业控制。

版权声明